

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- **BLACK BORDERS**
- **TEXT CUT OFF AT TOP, BOTTOM OR SIDES**
- **FADED TEXT**
- **ILLEGIBLE TEXT**
- **SKEWED/SLANTED IMAGES**
- **COLORLED PHOTOS**
- **BLACK OR VERY BLACK AND WHITE DARK PHOTOS**
- **GRAY SCALE DOCUMENTS**

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

(12) UK Patent Application (19) GB (11) 2 267 588 (13) A (43) Date of A publication 08.12.1993

(21) Application No 9212065.8

(22) Date of filing 06.06.1992

(71) Applicant
Motorola Inc

(Incorporated in the USA - Delaware)

Corporate Offices, 1303 East Algonquin Road,
Schaumburg, Illinois 60196, United States of America

(72) Inventors
Aviel Livay
Ricardo Berger
Alexander Joffe

(74) Agent and/or Address for Service
Sarah J Spaulding
Motorola, European Intellectual Property Operation,
Jays Close, Viables Industrial Estate, Basingstoke,
Hampshire, RG22 4PD, United Kingdom

(51) INT CL⁵
G06F 5/06

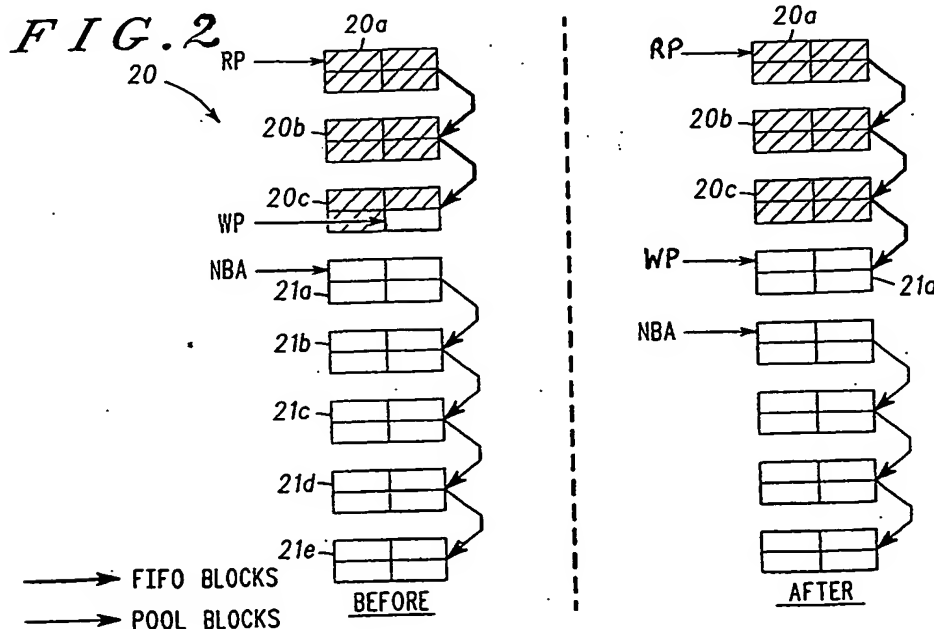
(52) UK CL (Edition L)
G4A AMT1
U1S S2202

(56) Documents cited
EP 0415862 A2

(58) Field of search
UK CL (Edition K) G4A AKB1 AMT1
INT CL⁵ G06F 5/06
Online database: WPI

(54) FIFO memory system

(57) A FIFO memory system 10 for handling transmission queues in a serial digital communication system comprises a plurality of blocks of memory 20a-c, 21a-e, blocks 20a-c being assigned a plurality of FIFO memories, and blocks 21a-e forming a pool of unassigned blocks of memory. A memory management means (LLT, PT) controls the addition of the unassigned blocks of memory from the pool to a FIFO memory on writing to the FIFO memory whereby the size of the FIFO memory is selectably variable, and the return of a block of memory from a FIFO memory to the pool once the contents of the block of memory have been read. In this way a FIFO has additional memory only when it needs it by dynamically applying small blocks of memory to the FIFO on request and, thus, overflowing and underrunning can be avoided whilst minimising the size of the FIFOs.



10

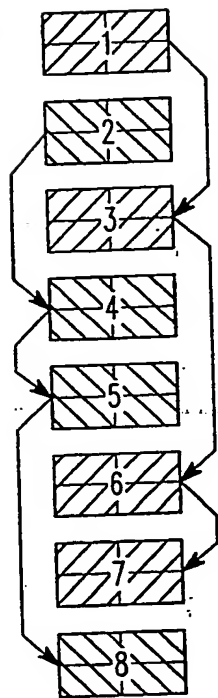


FIG. 1



 FIFO B
 FIFO A

FIG. 2

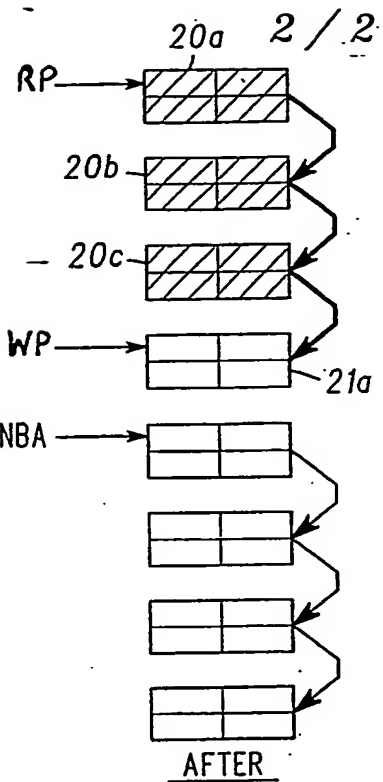
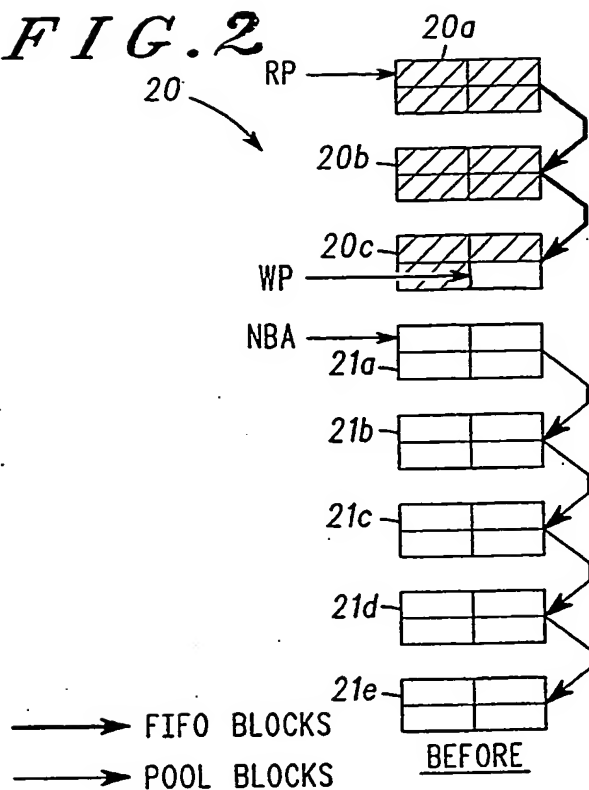
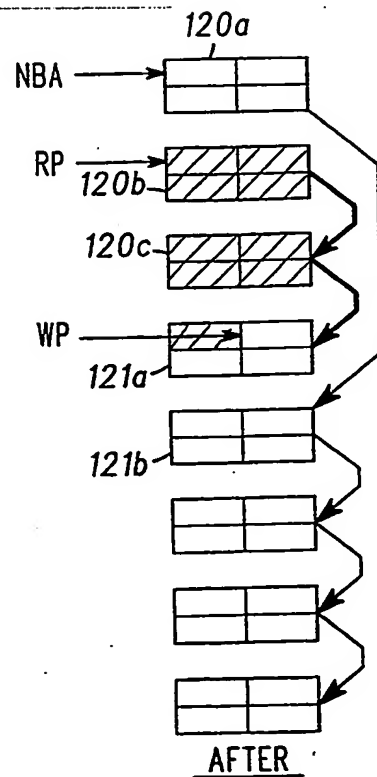
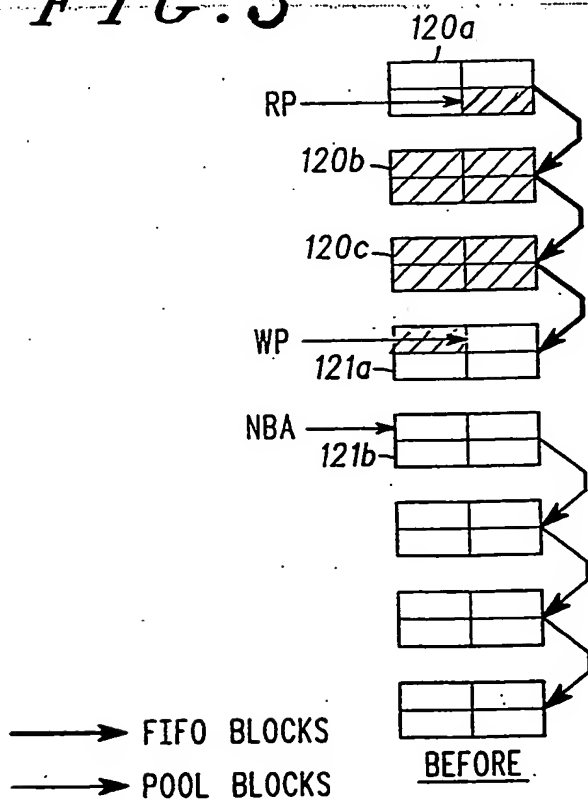


FIG. 3



FIFO Memory System

Background of the Invention

5 This invention relates to a FIFO memory system and more particularly to a FIFO memory system for use in a serial digital communication system.

With digital communication systems having two or more processes running concurrently, transmission queues are used in order
10 that the processes can efficiently transmit data to a communication line through a serial channel in the system. A common method of managing the transmission queues through the serial channel is to map each queue into one of a plurality of First-in-First-Out (FIFO) memories. The FIFOs are written to or filled by the system and
15 emptied or read from by the communication process or vice versa.

A problem with this method is implementing a plurality of FIFOs in a limited area. The preferred solution is to utilise RAM based FIFO memories since they appear to require the least area.

The filling rate of the FIFO (i.e. the rate at which data is written
20 to the FIFO) should normally be greater than the emptying rate of the FIFO (i.e. the rate at which data is read from the FIFO) onto the communication line. Typically, the FIFO issues Data Requests to the system any time a danger of underrun exists: underrun means carrying
out a read operation from the FIFO when it is empty.

25 The latency of the system bus carrying the data to be written must be considered in order to determine the minimum size of each FIFO. The latency of the bus is defined as the maximum period of time required by the system to supply the first data to the FIFO after a Data Request has been generated. For limited size memory FIFOs, the
30 maximum latency required is a critical parameter of the system configuration.

If L_s is the system latency having units of time, F_t is the rate at which the FIFO is emptied by the communication process and W_M is the minimum FIFO size below which Data Requests are generated and

assuming the FIFO is full when the first data is read by the communication process then,

$$WM = Ls \times Ft \quad (1)$$

5

With a FIFO having a size WM, once the FIFO is filled to its full size, the data requests will stop. However, Data Requests will be asserted again once the first data read is generated by the communication process in order to avoid an underrun state. Data Requests will therefore be
10 issued all the time.

In order to avoid this situation, a FIFO having a size WM + Delta must be implemented. When a Data Request is sensed by the system, the system will fill the FIFO to its maximum size (WM + Delta). However, the Data Request will be asserted again only after the FIFO is
15 emptied below the WM level.

A queuing system may comprise n different FIFOs for n different queues so that the total memory size MS must be greater than:

$$MS = (WM1 + Delta1) + (WM2 + Delta2) + \dots + (WMn + Deltan) \quad (2)$$

20

Assuming the FIFOs are n similar FIFOs and Delta = WM, equation 2 becomes

$$MS = 2 \times n \times WM$$

25

Therefore,

$$WM = \frac{MS}{2n} \quad (3)$$

30 From equation (1),

$$Ls \times Ft = \frac{MS}{2n} \quad (4)$$

And,

$$L_s = \frac{MS}{2n \times Ft} \quad (5)$$

Where L_s is the maximum latency acceptable to the memory system.

5 Thus, in order to account for the latency of the system bus and to avoid continuous Data Request being generated, the above solution requires additional FIFO memory: that is, $WM + \Delta$ for each FIFO. The serial communication process can access only one FIFO at a time and this one FIFO should fill up to the size $WM + \Delta$. Since the other
10 FIFOs also occupy $WM + \Delta$ of memory when they only require WM of memory, with the above solution there is $(n-1) \times \Delta$ of unused bytes of memory. Thus, large areas of memory are required but only a portion of the memory will be used at any one time.

15 Summary of the Invention

In accordance with the present invention there is provided a FIFO memory system comprising a plurality of FIFO memories for handling transmission queues in a serial digital communication system,
20 the memory system comprising:

a plurality of blocks of memory, each of the plurality of FIFO memories being assigned a block of the plurality of blocks of memory, the unassigned blocks of memory forming a block pool; and
memory management means for adding at least one of the
25 unassigned blocks of memory from the block pool to a FIFO memory on writing to the FIFO memory whereby the size of the FIFO memory is selectably variable, and for returning a block of memory from a FIFO memory to the block pool once the contents of the block of memory have been read.

30 Thus, in the FIFO memory system in accordance with the present invention, a FIFO has additional memory only when it needs it by dynamically applying small blocks of memory to the FIFOs on request.

The present invention therefore provides means by which the size of each of the plurality of FIFOs can be dynamically varied on

writing to the FIFO. An advantage of this arrangement is that the latency of the bus system is accounted for and the continual issuance of data requests is avoided but the memory area available is efficiently utilised: only the FIFO which is transmitting utilises WM + Delta of memory.

Brief Description of the Drawings

A FIFO memory system in accordance with the present invention will now be described by way of example only with reference to the accompanying drawings in which:

Figure 1 shows part of a FIFO memory system in accordance with the present invention;

Figure 2 shows part of the FIFO memory system in accordance with the present invention during a write operation; and

Figure 3 shows part of the FIFO memory system in accordance with the present invention during a read operation.

Detailed Description

Referring firstly to Figure 1, a FIFO memory system (only part 10 of which is shown in Figure 1) in accordance with the present invention comprises a plurality of FIFOs for handling the transmission queues in a digital communications system (not shown). The FIFO memory system 10 comprises a plurality of small blocks of memory, only eight 1-8 of which are shown in Figure 1.

Initially, each one of the plurality of FIFOs is assigned a small block of memory: for example FIFO A of Figure 1 is assigned block 2 and FIFO B is assigned block 1. In a FIFO memory system having n FIFOs, a predetermined number of the small blocks determined by n will be assigned to the plurality of FIFOs. The remaining blocks form a 'block pool' of memory from which all the plurality of FIFOs can 'borrow' when the FIFO requires additional memory.

As described in the introduction in order to avoid the continuous issuance of Data Requests during a read operation, the FIFO should

have a size of $WM + \Delta$. The present invention allows a FIFO to borrow the additional memory (Δ) from the 'block pool' during a write operation. For example, assuming in this case Δ is equal to WM , during a write operation a FIFO will increase its size to $2 \times WM$ by taking additional blocks from the 'block pool'. A FIFO that has been written to but is in a queue occupies memory having a size WM .

Blocks are returned to the 'block pool' when the block has been emptied during a read operation. The 'block pool' thus provides means by which the size of each of the plurality of FIFOs can be dynamically varied.

Preferably, each FIFO is implemented as a linked list of blocks in which each block in the list points to the next block of the FIFO. A Link List Table (LLT) contains the same number of entries as the number of blocks in the memory and each entry stores the address of the next linked block in memory. Table 1 represents the link list table for the eight blocks 1-8 shown in Figure 1. The lines on Figure 1 also represent which blocks are linked.

Block	Next linked block
1	3
2	4
3	6
4	5
5	8
6	7
7	x
8	x

Table 1

For each one of the plurality of FIFOs, a Read Pointer (RP) and a Write Pointer (WP) are defined. The location of the Read Pointer and Write Pointer are stored in a pointer table PT (not shown). The pointer table (PT) is updated depending on the contents of the link list table.

Each time a block of memory is written to or read from, the corresponding entry in the link list table is read so as to determine the address of the next linked block. The Read Pointer RP or Write Pointer WP is then re-defined according to the address of the next linked

5 block.

Read and write operations for one of the plurality of FIFOs in accordance with the present invention will now be described with reference to Figures 2 and 3.

A FIFO 20 is initially assigned a block 20a of memory. The Read and Write Pointers for FIFO 20 are defined according to the address of block 20a.

During a write operation, the FIFO block 20a is written to first and once this block has been filled the Write Pointer WP is updated so that it points to the next linked block whose address is stored in the entry for block 20a in the link list table. In the example shown in Figure 2, the next block is block 20b. Blocks 20b and 20c are written to in an identical manner. Memory blocks 21a-e form part of the block pool.

Once block 20c has been filled, FIFO 20 borrows a block from the block pool according to the entry for 20c in the link list table whereby the Write Pointer WP points to block 21a of the block pool.

Preferably, the block pool is also implemented as a linked list of blocks having a stack structure as shown in Figure 2. Thus, any block returned to the pool will be the first block available to a FIFO requiring it during a write operation. NBA (Next Block Available) indicates the top of the stack.

If FIFO 20 requires additional blocks, data will be written to blocks 21b-e in an order which depends on the link list table entries for these blocks.

Each one of the plurality of FIFOs uses a predetermined number of the small blocks of memory so as to occupy an area of memory having a size WM which is defined by the user. As discussed above the WM size determines the minimum level below which Data Requests are issued and depends on the latency of the system.

Referring now also to Figure 3 (like components to those of Figure 2 are referred to by the same reference numeral plus a hundred), during a read operation data is read from the blocks according to 16 where the Read Pointer is pointing. Data is thus read from block 20a and once this block has been emptied the Read Pointer RP is updated 10 so that it points to the next block of the FIFO according to the entry for block 20a in the link list table. In the example shown in Figure 3 the next block is block 20b. Data is then read from block 20b. Once block 20a has been emptied, the block becomes part of the block pool and is placed at the top of the stack as indicated by NBA. The entry in the 15 link list table for block 20a is then updated so that its next linked block is the next available block in the pool: that is block 21b. Thus, block 20a will be the first block from the block pool to be written to during a following write operation. 103d
19th

The invention recognises that the communication process can only read one FIFO at a specific time and so only one FIFO at any time requires additional memory of size Delta in order to avoid underrun. Thus, for n FIFOs, the total memory size MS is given by

20
$$MS = (WM1 + WM2 + WM3 \dots \dots \dots WMn) + Delta \quad (6)$$

Assuming the FIFOs are n similar FIFOs and Delta = WM, equation 6 becomes

25
$$MS = nWM + WM = WM (n + 1) \quad (7)$$

Substituting WM from equation 7 into equation 1 and rearranging gives

30
$$Ls = \frac{MS}{(n + 1) \times Ft} \quad (8)$$

Where Ls is the maximum latency acceptable to the memory system.

Thus, it is clear from comparing equations 8 and 5, for $n > 1$, that the maximum latency supported by the FIFO memory system in accordance with the present invention is bigger than the maximum latency supported by the conventional solution described in the introduction. From a different point of view, if the latencies of the systems L_s are the same, the total memory size required by the memory system in accordance with the present invention is reduced. Thus, the memory system in accordance with the present invention more efficiently utilises the memory available.

It will be appreciated that the present invention provides a FIFO memory system which optimises the latency of the bus system. Furthermore, the memory system in accordance with the invention can be relatively easily adapted to system buses having different latencies.

The plurality of FIFOs, the link list table and the pointer table are preferably all implemented in RAM.

Accessing a FIFO (i.e. a write or read operation) requires only one read from the link list table (LLT) and one write to the LLT. The same applies for the pointer table (PT). The LLT and PT are preferably implemented using dual ported RAMs whereby one read and one write can be done during the same memory cycle. This means that the preferred FIFO memory system is capable of updating the tables LLT and PT during the same cycle in which a FIFO is accessed. Thus, managing the FIFOs does not require any wait states and the memory system can be accessed each cycle.

A preferred embodiment of the present invention has been implemented in a FDDI (Fibre Distributed Data Interface) system interface. In this implementation, a memory array comprising 256 blocks of memory supported 30 FIFOs.

Claims

1. A FIFO memory system comprising a plurality of FIFO memories for handling transmission queues in a serial digital communication system, the memory system comprising:
 - a plurality of blocks of memory, each of the plurality of FIFO memories being assigned a block of the plurality of blocks of memory, the unassigned blocks of memory forming a block pool; and
 - memory management means for adding at least one of the unassigned blocks of memory from the block pool to a FIFO memory on writing to the FIFO memory whereby the size of the FIFO memory is selectably variable, and for returning a block of memory from a FIFO memory to the block pool once the contents of the block of memory have been read.
2. A FIFO memory system according to claim 1 wherein a FIFO memory occupies a predetermined number of blocks of memory after data is written to the FIFO memory, and wherein the memory management means on reading of the FIFO memory adds at least one of the unassigned blocks of memory from the block pool to the predetermined number of blocks of memory.
3. A FIFO memory system according to claim 1 or 2 comprising m linked blocks of memory and wherein the memory management means comprises a table having m entries, each of the m entries being associated with a respective one of the m blocks of memory and holding the address of the next block of memory to be linked to the respective one of the m blocks of memory.
4. A FIFO memory system according to any preceding claim wherein the system is implemented in RAM.
5. A FIFO memory system substantially as hereinbefore described with reference to the accompanying drawings.

10

Patents Act 1977
Examiner's report to the Comptroller under
Section 17 (The Search Report)

Application number

9212065.8.

Relevant Technical fields

(i) UK Cl (Edition K) G4A (AMTI, AKBI)

(ii) Int Cl (Edition 5) G06F 5/06

Databases (see over)

(i) UK Patent Office

(ii) ONLINE DATABASE: WPI

Search Examiner

MISS A C CLARKE

Date of Search

6 AUGUST 1992

Documents considered relevant following a search in respect of claims

1-4

Category (see over)	Identity of document and relevant passages	Relevant to claim(s)
X	EP 0415862 A2 (IBM) See column 2 line 40 - column 3 line 16	1-4

Category	Identity of document and relevant passages	Relevant to claim(s)

Categories of documents

X: Document indicating lack of novelty or of inventive step.

Y: Document indicating lack of inventive step if combined with one or more other documents of the same category.

A: Document indicating technological background and/or state of the art.

P: Document published on or after the declared priority date but before the filing date of the present application.

E: Patent document published on or after, but with priority date earlier than, the filing date of the present application.

&: Member of the same patent family, corresponding document.

Databases: The UK Patent Office database comprises classified collections of GB, EP, WO and US patent specifications as outlined periodically in the Official Journal (Patents). The on-line databases considered for search are also listed periodically in the Official Journal (Patents).